Predicted customer churn and evaluated model performance

Make predictions

Use the trained ANN model to predict customer churn based on the test dataset.

```
# Predict churn probabilities using the test data
churn_probabilities = ann_model.predict(X_test_scaled)
print(churn_probabilities)

# Convert probabilities into binary predictions (0 or 1) using a threshold of 0.5
churn_predictions = (churn_probabilities > 0.5).astype(int)
print(churn_predictions)
```

Running **print(churn_probabilities)**, we have below result: a 1D array of probabilities, one for each customer/row in the **X_test_scaled** dataset.

```
[[0.5281478 ]
 [0.08176879]
 [0.4478951 ]
 ...
 [0.3552407 ]
 [0.0285321 ]
 [0.00467736]]
```

Each value in the array represents the model's confidence that the corresponding customer belongs to the positive class (churn):

- Probabilities closer to 1: Higher likelihood of churn.
- Probabilities closer to 0: Lower likelihood of churn (no churn).

Based on the result above, we can interpret the result as:

- Customer 1: 52.81478% chance of churn
- Customer 2: 8.176879% chance of churn
- Customer 3: 44.78951% chance of churn
- And so on...

Later, we convert the probabilities into binary predictions (0 or 1) using a threshold of 0.5

- churn_probabilities > 0.5 → 1
- churn_probabilities < 0.5 → 0

Running **print(churn_predictions)**, we have below result:

```
[[1]
 [0]
 [0]
 ...
 [0]
 [0]
 [0]]
```

Evaluate model performance

To evaluate the model's performance, we analyse metrics like:

- Accuracy: fraction of correctly predicted samples.
- Precision: proportion of predicted churners that are actual churners.
- Recall: proportion of actual churners correctly identified.
- F1-Score: harmonic mean of precision and recall.
- Confusion matrix: breakdown of predictions into true positives, true negatives, false positives, and false negatives.

```python
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report
```

```python
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, churn_predictions)
precision = precision_score(y_test, churn_predictions)
recall = recall_score(y_test, churn_predictions)
f1 = f1_score(y_test, churn_predictions)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, churn_predictions)

# Display metrics
print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-Score: {f1:.4f}")

# Display confusion matrix
print("\nConfusion Matrix:")
print(conf_matrix)

# Generate a detailed classification report
print("\nClassification Report:")
print(classification_report(y_test, churn_predictions))
```

Result of running above codes:

```
Accuracy: 0.7918
Precision: 0.6471
Recall: 0.4770
F1-Score: 0.5491

Confusion Matrix:
[[923  96]
 [193 176]]

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.91      0.86      1019
           1       0.65      0.48      0.55       369

    accuracy                           0.79      1388
   macro avg       0.74      0.69      0.71      1388
weighted avg       0.78      0.79      0.78      1388
```

Result explanation

Accuracy: 0.7918 (approx. 79%)

- Overall, the model correctly classified ~79% of the samples.
- While accuracy is reasonable, it might not fully reflect model performance, especially for imbalanced datasets (e.g., fewer churners compared to non-churners).

Precision: 0.6471 (approx. 65%)

- Among the samples predicted as churners, ~65% are actual churners.
- Precision is moderate, the model is moderately good at avoiding false positives (predicting churn when it's actually no churn).
- If retention actions for false positives are costly, this precision might need improvement.

Recall: 0.4770 (~48%)

- The model identified ~48% of actual churners.
- Recall is relatively low, indicating the model misses more than half of the actual churners.
- Missing churners could be costly if the goal is to retain as many customers as possible.

F1-Score: 0.5491 (~55%)

- The F1-score balances precision and recall and gives an overall measure of the model's ability to predict churn.
- An F1-score of ~55% indicates room for improvement in both precision and recall.

Confusion matrix

```
Confusion Matrix:
[[923  96]
 [193 176]]
```

Prediction	Actual = No churn (0)	Actual = Churn (1)
Predicted = No churn (0)	True Negatives (TN): 923	False Negatives (FN): 193
Predicted = Churn (1)	False Positives (FP): 96	True Positives (TP): 176

- True Negatives (923):
  - The model correctly predicted 923 customers as non-churners.
- False Positives (96):
  - The model incorrectly predicted 96 non-churners as churners.
  - Moderate precision (65%) indicates some unnecessary retention actions.
  - Retention actions might be wasted on some non-churners, but the cost of false positives might be acceptable depending on business priorities.
- False Negatives (193):
  - The model missed 193 actual churners.
  - Low recall (~48%) shows many churners are not being identified.
  - Many churners are not being captured, which could hurt retention efforts.
- True Positives (176):
  - The model correctly predicted 176 churners.

Classification report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.91      0.86      1019
           1       0.65      0.48      0.55       369

    accuracy                           0.79      1388
   macro avg       0.74      0.69      0.71      1388
weighted avg       0.78      0.79      0.78      1388
```

- Class 0 (No Churn):
  - High precision (0.83) and recall (0.91) indicate the model is highly confident and accurate when predicting non-churners.
- Class 1 (Churn):
  - Precision is moderate (0.65), meaning ~65% of predicted churners are actual churners.

    - Recall is low (0.48), meaning the model misses more than half of the actual churners.

- Accuracy = 0.79 means approximately 79% of the test samples were classified correctly as either "churn" or "no churn"

- Macro avg: Precision = 0.74, Recall = 0.69, F1 = 0.71
    - Reflects the balanced performance across both classes.
    - Lower recall (0.48 for Class 1) reduces the macro avg recall and F1-score.

- Weighted avg: Precision = 0.78, Recall = 0.79, F1 = 0.78
    - Heavily influenced by Class 0 (majority class with higher performance metrics).
    - Suggests the model performs well overall but doesn't prioritize minority class performance.

From the numbers, we can have an insight of the model:

- Class imbalance: class 0 (No Churn) dominates the dataset, which skews accuracy and weighted averages in its favour.
- Low recall for class 1 (Churn): The model struggles to detect churners effectively, as indicated by the lower recall (0.48) and its impact on the macro avg.